

基于预测及蚁群算法的云计算资源调度策略

周文俊, 曹 健

(上海交通大学电子信息与电气工程学院, 上海 200240)

摘要: 研究云计算资源调度问题, 针对目前静态的网格资源调度算法只考虑任务完成时间最小化, 导致了不能满足动态的云计算资源调度要求。为了适应云计算的动态性和实时性, 解决云计算资源调度问题, 降低数据中心用电量, 提出一种基于预测及蚁群算法的云计算资源调度策略。当数据中心利用率较低时运行改进蚁群算法来合理调度虚拟机至宿主机, 通过动态趋势预测算法预测数据中心负载来智能开关宿主机。仿真结果表明, 采用预测及蚁群算法进行的云计算资源调度策略, 保证了云计算的实时性, 并有效减少数据中心用电量。

关键词: 云计算; 资源调度; 预测算法; 蚁群算法

中图分类号: TP311 文献标识码: B

Cloud Computing Resource Scheduling Strategy Based on Prediction and ACO Algorithm

ZHOU Wen-jun, CAO Jian

(Shanghai Jiao Tong University, School of Electronic Information and Electrical Engineering, Shanghai 200240, China)

ABSTRACT: Cloud computing resource scheduling was studied. The current static grid resource scheduling algorithms consider only the minimization of the makespan, so they can not meet the demands of cloud computing resource scheduling. In order to adapt to the dynamic and real-time nature and solve the issue of cloud computing resource scheduling and decrease the power consumption of datacenter, we proposed a cloud computing resource scheduling algorithm based on prediction and ACO algorithm. When the utility of datacenter is low, the improved ACO algorithm is executed to assign VMs to hosts. Dynamic tendency prediction strategy was used to predict the load of datacenter and turn on/off hosts. The results of simulation show that in the case of running VMs normally, cloud computing resource scheduling strategy based on prediction and ACO algorithm can reduce the power consumption of datacenter effectively.

KEYWORDS: Cloud computing; Resource scheduling; Prediction algorithm; ACO algorithm

1 引言

随着计算机性能的不断提高和互联网技术的飞速发展, 出现了云计算这种新兴的计算方式来满足人们日益增长的计算需求和海量数据处理需求。资源调度问题是云计算中一个核心问题。针对网格计算, 已经有许多静态资源调度算法被提出和应用^{[1][2][3]}。但是, 这些针对网格计算的静态资源调度算法并不适用于云计算的资源调度中, 原因包含以下三方面^[9]: 首先, 网格计算中的资源调度问题大多只考虑如何将一批网格任务合理分配到网格资源上使得当前资源得到最优分配。然而, 云计算中时刻存在着虚拟机的申请和释

放。所以, 静态资源调度算法不适用于云计算。其次, 网格计算中的网格任务运行时间是由其所选择的静态资源调度策略所决定的。但是, 云计算中虚拟机的使用时间是用户所决定的, 云计算资源调度策略不能决定虚拟机的使用时间。最后, 网格计算中的资源调度的目的是为了减少任务运行时间。然而, 云计算中虚拟机的使用时间是不能改变的, 所以, 云计算中资源调度的主要目的变成了如何降低数据中心的用电量。

针对上述情况, 提出一种结合预测算法与蚁群算法的云计算资源调度策略来降低数据中心的用电量。该资源调度策略有如下两个创新点: 第一、通过动态趋势预测算法预测未来短时间内数据中心负载的变化情况, 从而合理开关宿主机并且进行资源预留。预测数据中心负载的好处是提前判断下一时刻数据中心的负载情况, 避免频繁开关宿主机, 从

基金项目: 国家自然科学基金(61073021), 上海市科委项目
(10511501503, 10DZ1200200, 11511500102)

收稿日期: 2011-11-16 修回日期: 2011-11-30

而达到了资源预留的目的。第二、为了适应云计算的动态性和实时性 动态运行改进蚁群算法。在整个数据中心利用率较低时 利用改进蚁群算法将虚拟机集中到某些宿主机从而提高数据中心利用率。动态运行蚁群算法的好处是避免由频繁的虚拟机迁移导致数据中心产生额外用电量。

2 问题描述

假设一个数据中心由 n 台宿主机 $Host_1, Host_2, \dots, Host_n$ 组成 宿主机可提供资源为 $HostRe s_1, HostRe s_2, \dots, HostRe s_n$ 当前宿主机的空闲资源为 $FreeRe s_1, FreeRe s_2, \dots, FreeRe s_n$ 当前宿主的功耗为 $Power_1, Power_2, \dots, Power_n$ 当前宿主的利用率为 $Util_1, Util_2, \dots, Util_n$ 宿主机的最大功耗为 $MaxPower_1, MaxPower_2, \dots, MaxPower_n$ 宿主机的开机用电量为 $OnEnergy_1, OnEnergy_2, \dots, OnEnergy_n$ 在时间长度 t 中宿主机的用电量为 $Energy_1, Energy_2, \dots, Energy_n$ 。数据中心随时可能收到用户的虚拟机申请和释放请求 虚拟机随时会被分配或销毁。假设在时间长度 T 中有 m 个虚拟机请求到达数据中心 虚拟机所需资源为 $VMRe s_1, VMRe s_2, \dots, VMRe s_m$ 虚拟机的到达时间为 $Arrival_1, Arrival_2, \dots, Arrival_m$ 虚拟机的使用时间为 $Duration_1, Duration_2, \dots, Duration_m$ 。

云计算中的资源调度要解决的问题是在虚拟机的到达时间和使用时间未知的情况下 如何在 T 中对数据中心进行合理的资源调度和电源管理 从而使总用电量 $\sum_{i=1}^n Energy_i$ 最小化^[9]。由此可见 总用电量的高低成为了衡量一个云计算资源调度算法优劣的关键。

3 云计算资源调度算法模型

3.1 云计算资源调度系统原理

本文提出的云计算资源调度策略系统原理如图 1 所示。数据中心资源调度系统由虚拟机管理模块、虚拟机调度模块、宿主机电源管理模块和数据中心负载预测模块组成。系统拥有一个虚拟机队列和两个宿主机队列(开启/关闭) 分别维护和监控虚拟机和宿主机的状态。

1) 虚拟机管理模块

当用户向虚拟机管理器发送申请请求时 虚拟机管理器从开启宿主机队列中查找该虚拟机所需资源是否可以被满足 如果可以被满足 那么该虚拟机会被分配给此宿主机; 如果不可以被满足 那么从关闭宿主机队列中开启一个可以满足该虚拟机所需资源的宿主机 并且添加该虚拟机至虚拟机队列并更新宿主机队列。当用户向虚拟机管理器发送释放请求时 虚拟机管理器销毁该虚拟机 从虚拟机队列中删除该虚拟机并更新宿主机队列。

2) 虚拟机调度模块

监控开启宿主机队列的利用率 当开启宿主机队列的平均利用率 $\sum_{i=1}^n Util_i$ 低于所设定的利用率阈值时 使用改进的

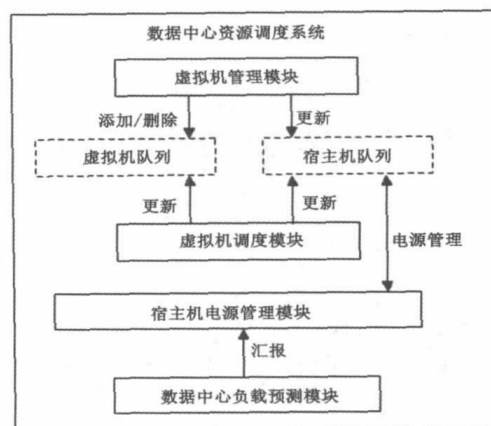


图1 数据中心资源调度系统原理

蚁群算法将所有正在运行的虚拟机重新分配到开启的宿主机上 从而尽可能使虚拟机集中在某些宿主机上运行 并且空出一些宿主机 使当前功率最低。这些空闲的宿主机将由宿主机电源管理模块决定是否将其关闭。

3) 数据中心负载预测模块

负责对数据中心的负载 $\sum_{i=1}^m VMRe s_i$ 做出短期预测 并把预测结果汇报给宿主机电源管理模块 提供宿主机开关的依据。设 t 时刻数据中心负载为 $VMRe sTotal_t$ 预测 $t+1$ 时刻的负载为 $VMRe sTotal_{t+1}$ 那么数据中心负载变化量 $\Delta VMRe sTotal$ 为 $VMRe sTotal_{t+1} - VMRe sTotal_t$ 。如果 $\Delta VMRe sTotal > 0$ 表明短期内申请虚拟机所需资源总和大于释放虚拟机所需资源总和; 如果 $\Delta VMRe sTotal < 0$ 说明短期内申请虚拟机所需资源总和小于释放虚拟机所需资源总和。

4) 宿主机电源管理模块

负责智能开关宿主机 达到云计算资源预留的目的。针对宿主机的智能开机 如果 $\Delta VMRe sTotal > \sum_{i=1}^n FreeRe s_i$ 从关闭宿主机队列中开启宿主机直到 $\Delta VMRe sTotal < \sum_{i=1}^n FreeRe s_i$ 这样提供了云计算资源预留 为短期内的虚拟机资源需求做好了前期准备。针对宿主机的关机 当宿主机 $Host_j$ 的利用率为零时 如果 $\Delta VMRe sTotal > \sum_{i=1}^n FreeRe s_i - FreeRe s_j$ 那么保留该宿主机; 否则 关闭该宿主机。

综上所述 数据中心资源管理系统的四个模块相互合作执行云计算资源调度和电源管理策略。虚拟机调度模块负责资源调度 宿主机电源管理模块和数据中心负载预测模块负责电源管理。虚拟机管理模块只负责虚拟机简单调度; 虚拟机调度模块负责虚拟机的调度 运行改进蚁群算法 并利用虚拟机迁移技术执行调度结果^[9]。电源管理方面 宿主机可以通过虚拟机管理模块和宿主机电源管理模块开启 通过宿主机电源管理模块关闭。

3.2 预测算法设计

采用基于文献[4]中的动态趋势预测策略,基本思想是按照当前时刻的负载趋势来预测下一时刻的负载。具体的工作流如图2所示^[9]。

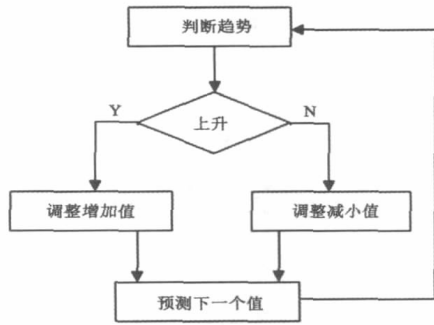


图2 动态趋势预测算法工作流

1) 判断负载趋势

比较当前时刻的负载情况和上一时刻的负载情况。如果当前时刻负载大于上一时刻负载,那么为上升趋势;如果当前时刻负载小于上一时刻负载,那么为下降趋势。

2a) 调整增加值

计算实际增加负载和历史平均负载。如果当前负载小于历史平均负载,调整增加值为:

$$IncVal' = IncVal + (Re_allIncVal - IncVal) \cdot AdaptDgr \quad (1)$$

其中 $IncVal$ 为当前增加值, $Re_allIncVal$ 为实际增加值;否则,调整增加值为:

$$IncVal' = \min(IncVal', IncVal \cdot PastGreater) \quad (2)$$

其中 $IncVal'$ 为式(1)的值, $PastGreater$ 为历史值中大于当前值的比例。

2b) 调整减小值

计算实际减小负载和历史平均负载。如果当前负载大于历史平均负载,调整减小值为:

$$DecVal' = V_T \cdot DecFct'$$

$$DecFct' = DecFct + (Re_allDecVal - DecVal) \cdot AdaptDgr \quad (3)$$

否则,调整减小值为式(4),其中 $PastLess$ 为历史值中小于当前值的比例:

$$DecVal' = V_T \cdot \min(DecFct', DecVal \cdot PastLess) \quad (4)$$

3) 预测下一个值

如果当前为上升趋势,那么预测下一时刻负载为当前负载加上增加值;如果当前为下降趋势,那么预测下一时刻负载为当前负载减去减小值。

调整增加值的实现由文献[4]提出,提出了调整减小值的实现。实验表明,该预测算法的预测误差率在15%以下。

3.3 改进蚁群算法设计

蚁群算法^[5]可以应用于组合爆炸问题,组合优化问题已经被证明是一个 $NP-hard$ 问题^[7]。蚁群算法最初是用来解

决旅行商问题的^[6]。蚁群算法的基本思想是模拟蚂蚁在寻找食物过程中释放信息素的行为来寻找组合爆炸问题的近似全局最优解^[9]。云计算中的资源调度策略对基本蚁群算法进行了改进,用来调度 m 个虚拟机到 n 个宿主机,通过虚拟机迁移技术使得在正常运行虚拟机的情况下数据中心的总功率最小。改进蚁群算法通过以下步骤解决该问题:

1) 初始化迭代次数 $t = 0$,为每个宿主机设置初始信息素 $C_i = HostRes_i / MaxPower_i$,即宿主机的“性价比”。

2) 每只蚂蚁拥有一个虚拟机队列 $VMList$,该虚拟机队列按资源需求量从大到小排列。

3) 蚂蚁根据式(5)轮盘赌选择宿主机。其中 P_j 为蚂蚁选择第 j 个宿主机的概率, τ_j 为第 j 个宿主机的信息素, η_j 为第 j 个宿主机的性价比。 α 和 β 分别表示信息素和性价比的重要程度, $allowed$ 为蚂蚁允许选择的宿主机集合,定义为不在 $tabuk$ 中并且 $HostRes$ 大于等于 $VMList$ 首元素需求量的宿主机。当蚂蚁选择宿主机 j 后,将宿主机 j 放入 $tabuk$,遍历 $VMList$ 将尽可能多的虚拟机调度到该宿主机,并在 $VMList$ 中删除这些虚拟机。

$$P_j(t) = \begin{cases} \frac{[\tau_j(t)]^\alpha \cdot [\eta_j]^\beta}{\sum_{k \in allowed} [\tau_k(t)]^\alpha \cdot [\eta_k]^\beta} & j \in allowed \\ 0 & otherwise \end{cases} \quad (5)$$

4) 如果 $VMList$ 不为空,返回执行3)。当所有蚂蚁的 $VMList$ 都为空时执行5)。

5) $t = t + 1$;

6) 所有宿主机按照式(6)来挥发信息素 ρ 为挥发率。

$$\tau_j(t) = (1 - \rho) \cdot \tau_j(t - 1) \quad (6)$$

7) 所有蚂蚁按照式(7)释放信息素, $\Delta\tau_j^k$ 为第 k 只蚂蚁在迭代次数 $t - 1$ 到 t 之间释放在宿主机 j 上的信息素。如果第 k 只蚂蚁在 $t - 1$ 到 t 之间经过宿主机 j ,那么 $\Delta\tau_j^k = Q / P_k$,其中 Q 为常量, P_k 为第 k 只蚂蚁经过宿主机的总功率;否则 $\Delta\tau_j^k = 0$ 。

$$\tau_j(t) = \tau_j(t) + \Delta\tau_j \quad (7)$$

$$\Delta\tau_j = \sum_{k=1}^M \Delta\tau_j^k$$

8) 选择本次循环中的最佳调度,即总功率最小调度。如果本次最佳调度的总功率比上次循环中的最佳调度小,那么替换最佳调度。

9) 判断结束条件。如果达到最大循环次数或者最佳调度的总功率已经超过一定循环次数不发生改变,那么结束蚁群算法;否则,重复式(2) - 式(8)。

4 仿真设计及结果分析

为了评估本文提出的云计算资源调度策略性能,选择 CloudSim^[8] 工具包进行了资源调度策略执行环境的模拟和算法实现的仿真。本文对 CloudSim 进行了扩展,并且改写了相关类和接口。

1) 实验条件

实验主要参数取值为: 时间长度 T 为 24 小时, 利用率阈值 $UtilThreshold$ 为 80%, 预测时间间隔为 5 分钟, 动态趋势预测算法中 $AdaptDgr$ 为 0.5, 改进蚁群算法中 α 为 1, β 为 5, ρ 为 0.5, 蚂蚁数量为 100, 最大迭代次数为 100。假定数据中心有 100 台宿主机, CPU 数在 [1, 4] 内随机产生, MIPS 为 1000, 虚拟机所需 CPU 数为 1, MIPS 在 [100, 900] 内随机产生, 运行时间在 5 分钟到 2 小时内随机产生。数据中心的负载情况如图 3 所示^[9]。

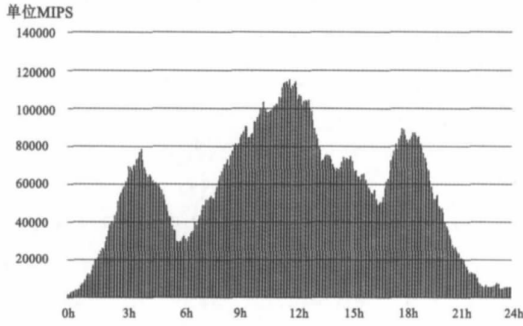


图 3 数据中心 24 小时负载情况

2) 实验结果及分析

为了验证算法的性能, 通过实验比较本文提出的云计算资源调度策略和以下四种资源调度策略^[9]:

策略 1: 当接受到虚拟机申请请求时, 遍历开启宿主机队列搜索满足虚拟机请求资源的宿主机, 若搜索失败, 那么继续在关闭宿主机队列中搜索, 直至搜索成功。监测宿主机, 当宿主机上无虚拟机运行时, 关闭该宿主机。

策略 2: 在策略 1 的基础上固定时间间隔运行改进后的蚁群算法, 运行时间间隔为 5 分钟。

策略 3: 在策略 1 的基础上固定时间间隔运行改进后的蚁群算法, 运行时间间隔为 30 分钟。

策略 4: 运行改进蚁群算法的时间间隔不固定, 当开启宿主机队列的平均利用率低于利用率阈值时运行改进蚁群算法。该策略不存在负载预测。

策略 5: 本文提出的云计算资源调度策略。

仿真比较五种资源调度策略在运行 24 小时后数据中心的用电量、虚拟机迁移次数和宿主机开关机次数。实验结果如表 1 所示。

表 1 五种资源调度策略仿真结果

策略	用电量(千瓦时)	迁移次数	开关机次数
策略 1	507.13	0	481
策略 2	451.98	26821	708
策略 3	502.42	4791	523
策略 4	430.57	6149	675
策略 5(本文)	406.17	6102	223

表 1 展示了五种不同资源调度策略的仿真结果。因为未使用改进蚁群算法, 策略 1 的用电量大于策略 2-5。虽然策略 2 和 3 都采用了改进蚁群算法, 但是策略 2 的用电量明显小于策略 3 的用电量, 所以, 调用改进蚁群算法的时间间隔是一个重要的参数。策略 2 和策略 3 的不足在于调用改进蚁群算法的时间间隔固定, 导致可能出现当宿主机利用率已经很高的时候仍然运行蚁群算法导致频繁开关机。因为策略 4 调用改进蚁群算法的时间间隔是动态的, 所以, 用电量比策略 2 和 3 低。CloudSim 仿真实验中, 策略 1-4 没有对整个数据中心进行有效的负载预测, 导致开关机次数过多反而增加了用电量。策略 5 利用动态趋势预测算法对数据中心的负载做出较准确的预测, 对空闲的宿主机进行了资源预留, 进而避免了频繁开关机带来的额外开销。综上所述, 本文提出的云计算资源调度策略有效减少了数据中心的用电量。

5 结束语

在云计算环境中, 资源调度是云计算中的核心问题, 云计算中资源调度策略的优劣直接影响了整个数据中心的性能和用电量。本文对云计算下的资源调度问题进行了深入研究, 提出了一种结合动态趋势预测算法和改进蚁群算法的云计算资源调度策略。本文的创新点为: 一、改进了蚁群算法, 并将改进蚁群算法应用到资源调度中, 利用虚拟机迁移进行了资源预留。二、利用动态趋势预测算法来预测数据中心负载变化, 并且提出了调整减小值的具体实现, 预测数据中心负载为电源管理提供了依据。实验结果表明, 本文提出的云计算资源调度策略有效降低了数据中心的用电量, 提高了数据中心的性能, 该策略具有很好的效果。

参考文献:

[1] 王刚, 钟志水, 黄永青. 基于蚁群遗传算法的网格资源调度研究[J]. 计算机仿真, 2009-4.

[2] 孔邵颖, 郭宏亮. 混合算法在网格任务调度中的应用研究[J]. 计算机仿真, 2011-9.

[3] 许元飞. 网格计算中任务调度算法的仿真研究[J]. 计算机仿真, 2011-8.

[4] Lingyun Yang, Ian Foster, Jennifer M Schopf. Homeostatic and Tendency-based CPU Load Predictions[C]. Proceedings of the 17th International Symposium on Parallel and Distributed Processing 2003.

[5] Marco Dorigo, Vittorio Maniezzo, Alberto Colomi. The Ant System: Optimization by a colony of cooperating agents[J]. Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.

[6] M Dorigo, L M Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. Evolutionary Computation, 1997, 1(1): 53-66.

(下转第 246 页)

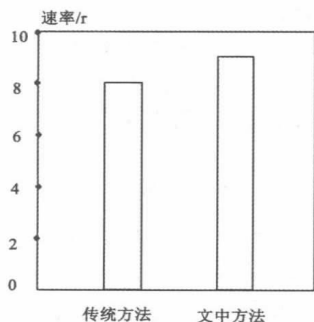


图3 两种方法拟合速率对比柱状图

实验数据对比表如表1示。

表1 实验数据对比表

拟合方法	拟合耗时/s	拟合速率/r
传统方法	10	8.2
文中方法	4	9.6

由上述实验结果可知,传统的基于遗传算法的数据点NURBS曲面拟合方法,由于拟合散乱数据点时需要多次进行遗传迭代运算,造成曲面拟合耗时长,单次耗时为10s,远高于文中提出的快速拟合方法,且拟合速率不高。而文中提出的基于蚁群的算法,模拟蚁群寻找食物的算法避免了多次的遗传迭代,大大缩短了拟合耗时,并将拟合速率提高到9.6,且此算法在曲面拟合完成后利用蚁群优化算法对曲面进行了优化,由图2可以看出,拟合出的曲面比传统方法拟合出的曲面更圆滑,满足了产品生产对图样的要求。因此,文中提出的方法能够有效提高散乱数据点进行NURBS曲面拟合的速率,并且具有较高的拟合质量和效率,取得了满意的结果。

5 结束语

文中提出了一种基于蚁群的数据点NURBS曲面拟合算法。通过采用蚁群寻址算法搜索出控制顶点和边界数据点集,计算曲面的权因子后完成NURBS曲面的拟合,并使用蚁群算法对拟合曲面进行优化,避免了传统方法多代遗传迭代造成的拟合速率不高的问题。实验表明,这种方法能够快速完成散乱数据点的NURBS曲面拟合,并且具有一定的拟合效率,取得了满意的结果,具有一定的使用价值。

参考文献:

- [1] Shewchuk, Jonathan Richard. Delaunay refinement algorithms for triangular mesh Generation [J]. Computational Geometry, 2002, 21(5): 21-74.
- [2] 孙玉文,贾振元,刘伟军,王越超. 基于自由曲面点云的快速原型制作技术研究[J]. 机械工程学报, 2003, 39(1): 56-59.
- [3] H Park, K Kim, S C Lee. A method for approximate NURBS curve compatibility based on multiple curves refitting [J]. Computer Aided Design, 2000, 32(2): 237-252.
- [4] 邱小湖,邱永成. 优化蚁群算法在无人机航路规划中的应用[J]. 计算机仿真, 2010, 27(9): 102-105.
- [5] 杜广东. NURBS技术剖析及其在船体线型表达上的行为研究[D]. 大连理工大学, 2000.

【作者简介】



张聚梅(1981-),女(汉族),山东无棣县人,硕士,助教,主要从事运筹与优化研究;

王洪伦(1981-),男(汉族),山东沾化县人,讲师,主要从事计算机技术研究;

张全信(1957-),男(汉族),山东沾化县人,教授,主要从事微分方程研究。

(上接第242页)

- [7] K W Tindell, A Burns, A J Wellings. Allocating hard real-time tasks: An NP-Hard problem made easy [J]. Real-time Systems, 1992, 4(2): 145-165.
- [8] R N Calheiros, R Ranjan, A Beloglazov, C A F De Rose and R Buyya. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms [J]. Software: Practice and Experience, 2011, 41: 23-50.
- [9] 周文俊. 集成测试管理平台仿真任务调度优化[D]. 上海交通大学, 2012年.

【作者简介】



周文俊(1986-),男(汉族),上海人,硕士研究生,主要研究方向:云计算;

曹健(1972-),男(汉族),江苏宜兴人,博士,教授,主要研究方向:服务计算、云计算。